REMARKS/ARGUMENTS

Favorable reconsideration of this application as presently amended and in light of the following discussion is respectfully requested.

Claims 1-24 are presently active in this case. Claims 1, 8 and 14 having been amended and Claims 22-24 added by way of the present Amendment.

In the outstanding Office Action stand rejected under 35 U.S.C. § 102(e) as anticipated by Hauser et al. (U.S. Patent 5,850,395, hereinafter called "Hauser"). Claims 19-21 were objected to as being dependent upon a rejected base claim, but otherwise indicated as being allowable if rewritten in independent form.

Applicants acknowledge with appreciation the indication that Claims 19-21 include allowable subject matter. However, since Applicants consider that Claims 1, 8 and 14 define allowable subject matter, Claims 19-21 have presently been maintained in dependent form.

Applicant further acknowledges with appreciation the courtesy of an interview granted to Applicant's attorney on June 8, 2004, at which time the outstanding issues in this case were discussed. During the interview, Applicant's attorney explained the operation of the invention and the differences of the claimed invention over Hauser as hereinafter described. However no agreement was reached pending the Examiner's detailed reconsideration of the application upon formal submission of a response.

Applicant respectfully traverses the outstanding rejection on the merits, because in Applicant's view the claimed invention clearly is not taught by Hauser.

In particular, Claim 1 defines a queue control device including first and storage areas and a controller operatively connected thereto. The first storage area is configured to store first and second queues respectively storing first and second elements each having an address specifying a next element, wherein a head address of the second queue is stored in a tail element of the first element and a tail address of the first queue is stored in a tail element of the second element. The second storage area stores as a first pointer information a head address specifying a head element of the first element included in the first queue and a tail address specifying a tail element of the second element included in the second queue as a second pointer information. Thus, as explained in the Applicant's prior amendment filed January 2, 2004, the claimed invention is directed to plural queues that are connected and controlled as a single queue, thereby preventing storage capacity from increasing.

In contrast to Applicant's claimed invention, Hauser does not teach a system or technique in which plural queues are connectively controlled as a single queue. On the contrary, in Figure 6 of Hauser queue pointers and respective descriptors for queues 1-3 are illustrated. Queues 1-3 are independent of each other, and independently accessed, and independently controlled. This is believed to be completely evident from the "queue descriptors" and "logical queues" illustrated at the bottom of Figure 6. For example, note that the Queue 1 Descriptor indicates the cell number 11 as its head and the cell number 14 as its tail and as is evident from Figure 6, no connectivity to other queue pointers are indicated in the Hauser et al Queue Descriptors.

In fact, each queue descriptor shown in FIG. 6 of Hauser et al. manages only one type of queue, and not queues of different types. Claims 1, 8 and 14 recite that a storage area manages two types of queues by managing the head address of a first queue and the tail address of a second queue. In other words, two types of queues can be managed by managing one head address and one tail address. Such a structure/operation achieves the advantage of reducing storage capacity of the second storage area. However, Hauser et al. does not suggest the technical concept of managing two types of queues by managing the head address of a first queue and the tail address of a second queue. In addition, Hauser et al.

does not at all suggest the technical concept of managing plural queues for reducing storage capacity of the queue descriptor. With Hauser et al., two head addresses and two tail addresses must be stored in the queue descriptor in order to manage two types of queues. Thus, the storage capacity of the queue descriptor must be twice the size of the storage capacity of claims 1, 8 and 14.

Claims 1 and 8 recite that two types of queues are managed by a second storage area storing one head address and one tail address, and that a controller processes the queues as two different types of queues. On the other hand, Hauser et al. does not suggest such a controller.

With the configuration shown in FIG. 6 of Hauser et al., queue 1 is processed using the queue 1 descriptor, and queue 2 is processed thereafter, by referring to the content of the queue 2 descriptor. Thus, with Hauser et al., when processing of one queue is finished, the content of the subsequent queue descriptor must be referred to.

On the other hand, with claims 1, 8 and 14, the head address of the first queue and the tail address of the second queue are set in the second storage area (storage area). Further, the head address of the second queue is set in the tail element of the first queue, and the tail address of the first queue is set in the tail element of the second queue. Accordingly, unlike with Hauser et al., the content of the queue descriptor does not need to be referred to when processing of the first queue is finished. In other words, the first and second queues can be successively processed, and processing can be completed at high speed.

At page 3 of the outstanding Office Action, it is stated,

> "In Hauser et al., one queue number is able to point to a second queue number, and the list descriptor holds the queue numbers of the first entry, or head, of the list and the last entry, or tail, of the list. The queue number of the head of the list is used to index into the list pointers and read the queue number of the

> second entry in the list. Likewise, the queue number of the second entry is used to index into the list pointers and read the queue number of the third entry, and so on until the queue number read is equal to the queue number of the last entry, or tail of the list. Therefore, it is clear that in Hauser et al., in order to add a queue to a list, the queue number of the queue being added to the list is written to the list pointer location indexed by the present tail of the list. That is, when queue 2 is added for immediate transmission to the list, the address of the head of queue 2 is written to the present tail of queue 1. ..."[1]

However, the above description in the outstanding Office Action appears to be heavily influenced by a hindsight reconstruction of Applicant's claimed invention, and it is respectfully submitted is not an accurate representation of the actual teachings of Hauser provided at, for example, column 14, line 52 – column 15, line 3. There, Hauser describes the structure of "lists" comprised of individual queues, wherein each list has a list number and each queue has a queue number. As explained by Hauser,

> ...lists behave much as queues, except that a queue is a linked list of cell numbers while a list is a linked list of queue numbers. With queues, cell numbers are serviced in order, i.e., the first cell number added to a queue is the first removed. With lists, queue numbers are serviced in order, i.e., the first queue number added to the list is a first removed. Lists are implemented as a linked list of queue numbers. Each queue number on the list points to the next queue number on the list using the queue number itself as a pointer. Each list has a separate structure, called the list descriptor, maintained internal to the FSPP unit 134 to point to the head and tail of the list. ...

Thus, Hauser clearly indicates that queues are processed by queue numbers in a list, and the individual queues themselves are not per se linked, only the queue numbers are linked. Thus, the tail cell in an individual queue has no relation to the head or tail of a subsequent queue, since that information is independently processed by means of the queue numbers and not pointers included with queue cells.

[1] Outstanding Official Action, at page 3, lines 6-17.

In short, it is respectfully submitted that the prior art does not teach the first and second storage areas and controller by which the first and second storage areas are operatively connected, as claimed, and does not teach the tail element of a first queue pointing to the head element of a second queue and the tail element of the second queue pointing to the tail element of the first queue, as claimed. For these reasons, it is respectfully submitted that Hauser in no way anticipates or obviates the claimed invention, and that the pending claims patentably define over Hauser et al.

New Claims 22 and 23 recite that the second storage area does not store the tail address of a first queue and the head address of a second queue. Further, new Claim 24 recites that no tail address in the first queue is stored in the storage area no head address in the second queue is stored in the storage area. Accordingly, with the claimed invention, it is possible to reduce the storage capacities of the second storage area (Claims 1 and 8) and the storage area (Claim 14). On the other hand, FIG. 6 of Hauser et al. show a queue 1 descriptor and a queue 2 descriptor which include both the head and tail addresses of a first queue and both the head and tail addresses of a second queue. Thus, unlike claims 22 to 24 of the present application, Hauser et al. do not teach or suggest reducing the required storage capacity.

Consequently, in view of above comments, it is respectfully submitted Claims 1-24 are in condition for allowance, and an early and favorable action to that effect is respectfully requested.

Respectfully submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.

Eckhard H. Kuesters
Attorney of Record
Registration No. 28,870

22850

Tel: (703) 413-3000
Fax: (703) 413 -2220

I:\ATTY\EHK\AMEND-RESPONSES\0039\19S\197336US-AMT FILED JUNE 23.DOC